

**IN THE CLAIMS:**

Please amend the claims as follows:

1. (Currently Amended): A method computer readable medium storing a sequence of instructions for execution on a computer for obfuscating code, comprising:
  - obtaining an algorithm instantiated in code that includes a sequence of instructions operating on operands stored in registers;
  - obtaining at least two instructions of the code; and
  - inserting inert executable instructions between the at least two instructions obtained, wherein one or more of the inert executable instructions temporarily modify one of the operands stored in one of the registers and another one or more of the inert executable instructions restore the one operand in the one register.
2. (Currently Amended): The method of claim 1, wherein the inert executable instructions include one or more inert unity instructions.
3. (Currently Amended): The method of claim 1, wherein the inert executable instructions include one or more inert logic instructions.
4. (Currently Amended): The method of claim 1, wherein the inert executable instructions include one or more inert branch instructions.
5. (Currently Amended): The method of claim 1, further comprising reordering and intermingling one or more of the inert executable instructions and one or more of the instructions in ~~within~~ the code.
6. (Cancelled)
7. (Currently Amended): The method of claim 1, further comprising randomly or pseudo-randomly obtaining the inert executable instructions from a pool of instructions.

8 – 9 (Cancelled)

10. (Original): The method of claim 1, further comprising obtaining a seed from a pool of seeds.

11. (Currently Amended): The method of claim 10, wherein the seed is randomly or pseudo-randomly selected from a pool of seeds.

12. (Cancelled)

13. (Currently Amended): The method of claim 10, wherein the seed is used to determine locations in the code for insertion of the ~~insert~~ inert executable instructions.

14. (Currently Amended): The method of claim 10, wherein the seed is used to select a number of inert executable instructions for insertion.

15. (Original): The method of claim 10, wherein the seed is used to set a threshold level for obfuscation.

16. (Original): The method of claim 10, wherein a seed is selected for each build of the code.

17. (Currently Amended): A programmed computer, comprising:

a central processing unit configured to execute a code obfuscation program to insert inert instructions in portions of one or more programs;

an input/output interface configured to interface with the central processing unit;  
and

a memory, the memory storing at least a portion of the code obfuscation program the program comprising a sequence of instructions for:

obtaining an algorithm instantiated in code that includes a sequence of algorithm instructions wherein operating on operands stored in registers;  
obtaining at least two instructions of the code; and  
inserting executable inert instructions between the at least two instructions obtained, wherein one or more of the inert executable instructions temporarily modify one of the operands stored in one of the registers and another one or more of the inert executable instructions restore the one operand in the one register.

18. (Original): The programmed computer of claim 17, wherein the input/output interface is coupled to the memory.

19. (Original): The programmed computer of claim 18, wherein the memory is coupled to the central processing unit.

20. (Original): The programmed computer of claim 18, wherein the memory stores at least one seed used during execution of the code obfuscation program.

21. (New): A computer readable medium as claimed in claim 1, wherein the instructions are inserted only between the at least two instructions which are determined to be non time dependent or time sensitive.

22. (New): A computer readable medium as claimed in claim 1, wherein a sequence of the inert executable instructions is dispersed throughout the sequence of algorithm instructions.

23. (New): A computer as claimed in claim 17, wherein the inert executable instructions are inserted only between the at least two instructions which are determined to be non time dependent or time sensitive.

24. (New): A computer as claimed in claim 17, wherein a sequence of the inert executable instructions is dispersed throughout the sequence of algorithm instructions.